

IN THE UNITED STATES DISTRICT COURT
FOR THE DISTRICT OF MASSACHUSETTS

NETVIEW TECHNOLOGIES, INC.

Plaintiff and
Counterclaim-Defendant,

v.

MICROSOFT CORPORATION

Defendant and
Counterclaim-Plaintiff.

CIVIL ACTION NO. 09 cv 12072 DPW

**DEFENDANT MICOROSOFT CORPORATION'S OPENING BRIEF
IN SUPPORT OF ITS PROPOSED CLAIM CONSTRUCTIONS**

TABLE OF CONTENTS

	Page
I. INTRODUCTION	1
II. BACKGROUND	1
A. Using Spreadsheets As Mathematical Functions	1
B. '776 Patent Claim 1	3
C. The Patent-In-Suit	4
III. CLAIM CONSTRUCTION	6
A. The Law Of Claim Construction.....	6
B. Proposed Constructions Of Asserted Claims 1 And 10.....	7
1. “spreadsheet”/“the spreadsheet”	7
2. “defining a parameter external to the spreadsheet”	10
3. “design time”/“run time”	15
4. “a parameterized workbook”/“associating the parameter with the spreadsheet at design time to define a parameterized workbook”	17
5. “reference the parameter”/“the parameter applies to the spreadsheet as a whole, thereby allowing any formula in the spreadsheet to reference the parameter”/“defining a formula within the spreadsheet at design time using (<i>sic</i>) parameter”	21
6. “receiving a value for the parameter at run time”	24
7. “rendering an output based on computed cell values”	25
8. Claims As A Whole	26
IV. SYSTEMATIC ERRORS IN PLAINTIFF’S PROPOSED CONSTRUCTIONS	28
V. THE CLAIMS ARE INDEFINITE	29

TABLE OF AUTHORITIES

	Page
Cases	
<i>Akamai Techs., Inc. v. Limelight Networks, Inc.</i> , 2010 WL 5151337 (Fed. Cir. Dec. 20, 2010)	11, 12
<i>Astrazeneca LP v. Apotex, Inc.</i> , 2010 WL 4286284 (Fed. Cir. Nov. 1, 2010).....	27
<i>Baldwin Graphic Sys., Inc. v. Siebert, Inc.</i> , 512 F.3d 1338 (Fed. Cir. 2008).....	10
<i>Bicon, Inc. v. The Straumann Co.</i> , 441 F.3d 945 (Fed. Cir. 2006).....	21
<i>Exxon Research & Eng'g Co. v. United States</i> , 265 F.3d 1371 (Fed. Cir. 2001).....	31
<i>Hewlett-Packard Co. v. Mustek Systems, Inc.</i> , 340 F.3d 1314 (Fed. Cir. 2003).....	7
<i>In re Nehls</i> , 2008 WL 258370 (B.P.A.I. Jan. 28, 2008)	27
<i>J.T. Eaton & Co. v. Atlantic Paste & Glue Co.</i> , 106 F.3d 1563 (Fed. Cir. 1997).....	30
<i>O.I. Corp. v. Tekmar Co.</i> , 115 F.3d 1576 (Fed. Cir. 1997).....	12
<i>Ormco Corp. v. Align Tech., Inc.</i> , 498 F.3d 1307 (Fed. Cir. 2007).....	25, 28
<i>Phillips v. AWH Corp.</i> , 415 F.3d 1303 (Fed. Cir. 2005) (<i>en banc</i>)	6
<i>Power Mosfet Technologies, L.L.C. v. Siemens AG</i> , 378 F.3d 1396 (Fed. Cir. 2004).....	6, 7
<i>SmithKline Beecham Corp. v. Apotex Corp.</i> , 403 F.3d 1331 (Fed. Cir. 2005).....	6, 29
<i>United Carbon Co. v. Binney & Smith Co.</i> , 317 U.S. 228 (1942).....	30
Statutes	
35 U.S.C. § 112.....	17, 30, 31

I. INTRODUCTION

Three colleagues at Parametric Technology Corporation (PTC) decided that despite having signed agreements assigning inventions to PTC, they would try to independently commercialize an idea they had about the use of spreadsheets. Although the idea was in reality an old one, they were able to obtain a patent, in part by not telling the Patent Office what they knew about the prior art. Their business endeavor went nowhere, and they gave up after a few years. Now, the remaining shell of their company, NetView, asserts that Microsoft has infringed some claims of their patent. NetView makes this accusation even though it is not the true owner of the patent, and thus lacks standing, and even though the accused software uses a technique that the patent itself distinguishes from its alleged “invention.”

With these and other issues in the background, the parties dispute most of the language of the asserted patent claims.

II. BACKGROUND

A. Using Spreadsheets As Mathematical Functions

The patent-in-suit involves the use of a “spreadsheet” (or “worksheet”); that is, a logical grid of cells containing formulas and data. The patent’s basic concept is to use a spreadsheet in performing the mathematical function of computing a result from a variable input “parameter.” Both the input parameter and the output result are conceived as being **external** to the spreadsheet.

The patent attempts to distinguish this idea from the conventional “what-if scenario” use of a spreadsheet, in which a user can change—that is, provide a new data value for—one or more of the spreadsheet’s cells, in order to compute and view a new result based on those new input values. For example, what if we raised salaries by 10%, how would that affect our annual profit? The patent says, “This [conventional] mechanism allows a spreadsheet user to perform what-if

scenarios only by **modifying cell values** and viewing or saving the effects of the changes. . . . Prior art spreadsheet programs make the user **vary cell values** if the user wants to perform different calculations.” (’776 Patent at 1:39-2:11 (emphases added)). (This is also how the accused technique works.) In contrast, the patent proclaims, with its “parameterized workbooks[], a user does not have to vary cell values in order to perform different calculations, including what-if scenarios.” (*Id.* at 11:45-47). Instead, in the patent, the parameter is external to (independent of) the spreadsheet, and thus is not a cell of the spreadsheet. Because the parameter and the output are each external, their definitions can be changed without changing the spreadsheet itself. Thus, the spreadsheet can be reused with different parameter and/or output definitions, making the “parameterized workbook” a “reusable **computational** building block” (*id.* at 9:12-13 (emphasis added)), according to the patent.

But, the patent’s idea of associating a spreadsheet with an **external** parameter and **external** output is an old one, described in multiple prior-art references that were not disclosed to or considered by the Patent Office when it granted the patent. (*See* Dkt. No. 39 at 12-16 and, e.g., Exs.¹ A-D). For example, a 1997 paper (Ex. A) by Chris Clack and Lee Braine of University College London, entitled “Object-Oriented Functional Spreadsheets,” described the same concept five years earlier using strikingly similar language as the patent-in-suit. The paper—not considered by the Patent Office—described a “reusable” “parameterized worksheet” as constituting a “function” that would take an input “parameter” and return a “result”:

A workbook has been defined as a collection of worksheets; we now allow one or more of these worksheets to take parameters and to return a result. Thus, the spreadsheet application is split into two parts: (i) simple worksheets that do not take parameters (and which provide a traditional non-reusable grid), and (ii) worksheets that take one or more parameters and return a single result (which provide reusable “worksheet templates”). A parameterized worksheet is thus a

¹ All Exhibits are attached to the accompanying Declaration of Mung Wah Tam.

function definition. . . . We believe that the grid format provides a useful structuring mechanism for the function body, wherein cells may contain expressions with references to the worksheet's parameters. (Ex. A at MS-NV118405).

B. '776 Patent Claim 1

The '776 Patent's Claim 1 is directed to this same idea of defining a "parameterized workbook" function and using it to compute an output from an input value of an external parameter:

1. A method for utilizing a spreadsheet, the method comprising:

defining a parameter external to the spreadsheet;

associating the parameter with the spreadsheet at design time to define a parameterized workbook, wherein the parameter applies to the spreadsheet as a whole, thereby allowing any formula in the spreadsheet to reference the parameter;

receiving a value for the parameter at run time;

computing cell values in the spreadsheet that are dependent, directly or indirectly, on parameter;

and rendering an output based on the computed cell values. ('776 Patent at 26:12-23).

Claim 10 depends from (and thus incorporates) Claim 1, and adds a single step: "10. The method of claim 1, further comprising: defining a formula within the spreadsheet at design time using the parameter." (*Id.* at 26:55-57).

Neither Claim 1 nor Claim 10 restricts who or what may perform any of its steps, or how the steps are performed. The claims do not specify who or what must "define" the external parameter, or who or what is to "associate" it with the spreadsheet, or how this is to be done. Nor do the claims specify who or what is to "receive" a parameter value, or "compute" cell values, or "render" an output, or how any of this is done. For example, the claims do not recite the use of any kind of "software," "program," "computer," "memory," or "processor." The

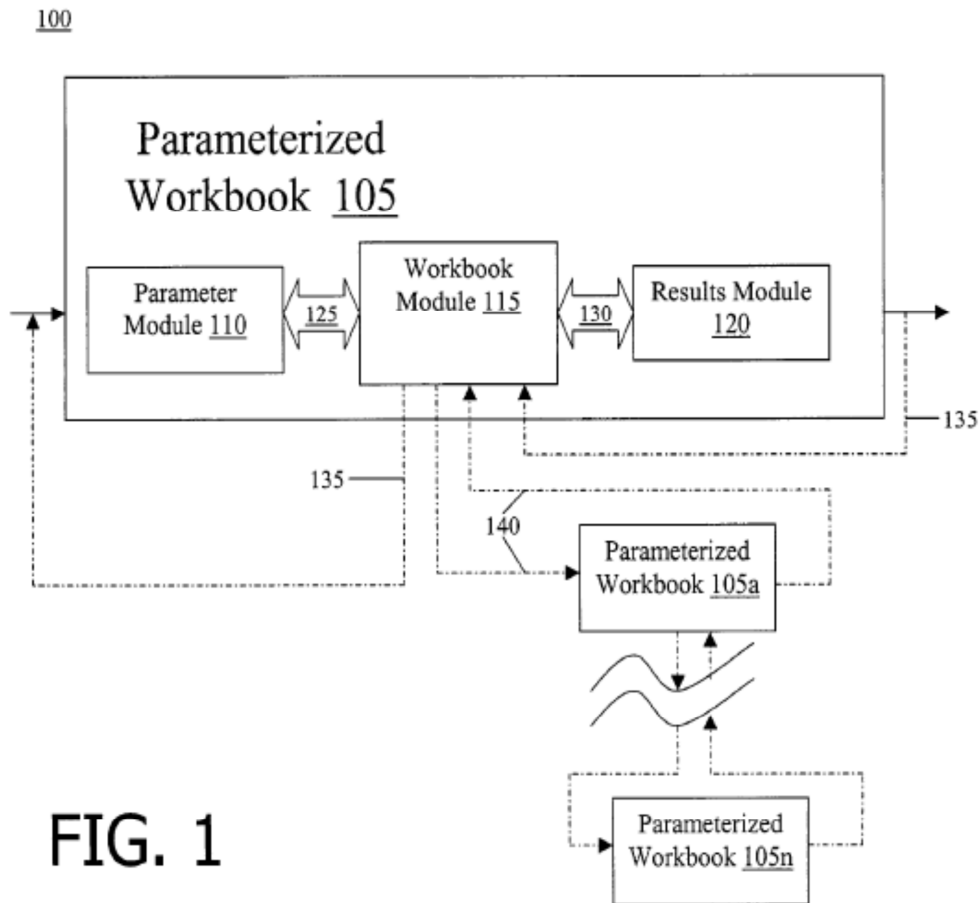
claims also do not require any particular application of their mathematical operations, or operation on any particular articles. Instead, the claims are directed to pure math, not a particular physical-world implementation of that math.

C. The Patent-In-Suit

Math pervades this patent. Variants of the word “calculate” appear at least 143 times in the patent, “formula” at least 81 times, and “data” at least 229 times. (Dkt. No. 30 at ¶ 8). The patent refers to its spreadsheet-workbook “function” as a “reusable **computational** building block” (’776 Patent at 9:12-13 (emphasis added)), and as including “any **logical unit of spreadsheet computation**, comprising one or more spreadsheets and associated **formulas**” (*id.* at 11:39-41 (emphases added)).² Similarly, the patent states that “the parameterized workbook 105a is a **function mapping a set of inputs 110 to a set of outputs 120.**” (*Id.* at 13:19-21 (emphases added)).

As illustrated in Fig. 1 (reproduced below) of the patent’s application as filed (Ex. E), the patent’s “parameterized workbook” function has three separate but associated (logical) modules: a parameter module (defining an input parameter), a workbook module (defining the spreadsheet calculation logic), and a results module (defining the output). The arrows 125 and 130 represent an “association” of these modules, to form the parameterized workbook. (’776 Patent at 8:45-67, 10:35-11:18).

² “Logical” refers to an abstract, conceptual design or idea, not a particular physical entity. (E.g., ex. G (Microsoft Computer Dictionary (4th ed. 1999) at 272).



Although the three modules may be implemented using computer software, the patent states that they need not be implemented using software. ('776 Patent at 8:63-67).

The three-module outline of Fig. 1 applies to any mathematical function, such as a simple equation. A simple equation has an input variable, an output result, and a mathematical operator that “maps” the former to the latter. For example, the mathematical equation for calculating the area of a circle, $A = \pi r^2$, applies to the input variable r the mathematical operators of squaring, and multiplication by the constant π , to provide the result A . The variable r is defined, independently of these mathematical operators, as the length of the radius of a circle in some dimension (e.g., feet), and the output A is defined, again independently of these mathematical operators, as the area of the circle in some dimension (e.g., square feet). Inputting a value 2 for

variable r outputs the value 12.56 for A , while inputting a value 4 results just as necessarily in an output of the value 50.24 for A .

Claim 1 of the patent in suit recites defining such a three-module mathematical function (the “parameterized workbook”) and then operating it on a variable input value (parameter) to render an output value.

III. CLAIM CONSTRUCTION

A. The Law Of Claim Construction

“The claims of a patent define the invention to which the patentee is entitled the right to exclude.” *Phillips v. AWH Corp.*, 415 F.3d 1303, 1312 (Fed. Cir. 2005) (*en banc*). Claim language generally is given the meaning it would have to one of ordinary skill in the relevant art, at the time the application was filed, in view of the patent specification. *Id.* at 1313. “Claims must be read in view of the specification, of which they are a part.” *Id.* at 1315 (citation omitted). The patent’s specification “is always highly relevant to the claim construction analysis. Usually, it is dispositive; it is the single best guide to the meaning of a disputed term.” *Id.* (citation omitted).

When construing claim language, a court cannot rewrite the claims, nor can the court add or subtract words from the claims. *See SmithKline Beecham Corp. v. Apotex Corp.*, 403 F.3d 1331, 1339-40 (Fed. Cir. 2005).

Most *Markman* claim-construction proceedings focus on claim terms in isolation. But the Federal Circuit has warned that sometimes it is important to construe a patent claim as a whole. *See, e.g., Power Mosfet Technologies, L.L.C. v. Siemens AG*, 378 F.3d 1396, 1404 (Fed. Cir. 2004) (“The terms in the Special Master Report were construed in isolation, and at no other time did the district court or the Special Master construe the claims as a whole.”); *id.* at 1410 (This “limited construction left substantial ambiguity as to the meaning of the claims as a

whole....”); *id.* at 1412 (“[A] construction of the claims as a whole would have been beneficial to the litigants.”). In this case, Microsoft has proposed such “as a whole” constructions; NetView has not.

Finally, one purpose of the claim-construction process is to enable the crafting of jury instructions whose words may be understood in their ordinary sense by lay jurors. *See Hewlett-Packard Co. v. Mustek Systems, Inc.*, 340 F.3d 1314, 1320-21 (Fed. Cir. 2003). Therefore, any construction should strive to use language understandable to a lay jury, rather than language having special meaning as terms of art. By the same token, the construction process is designed to eliminate the prospect of the parties and their expert witnesses still disputing constructions before the jury at the time of trial.

B. Proposed Constructions Of Asserted Claims 1 And 10

1. “spreadsheet”/“the spreadsheet”

<u>Context of Terms Used In Claim</u>	<u>Construction</u>
1. A method for utilizing a spreadsheet , the method comprising: defining a parameter external to the spreadsheet ; associating the parameter with the spreadsheet at design time to define a parameterized workbook, wherein the parameter applies to the spreadsheet as a whole, thereby allowing any formula in the spreadsheet to reference the parameter; . . . computing cell values in the spreadsheet that are dependent, directly or indirectly, on parameter; . . .	<p><u>“spreadsheet”</u>: A single two-dimensional grid of cells forming a logical unit of calculation. Each cell can contain a formula for calculating a data value from another value. Each cell not containing a formula can instead contain a data value.</p> <p><u>“the spreadsheet”</u>: The same spreadsheet that is referenced as “the spreadsheet” elsewhere in the claim, not a different copy of it.</p>

“spreadsheet”: A single two-dimensional grid of cells forming a logical unit of calculation. Each cell can contain a formula for calculating a data value from another value. Each cell not containing a formula can instead contain a data value.

Single Two-Dimensional Grid of Cells: The ’776 Patent defines “spreadsheet” as a single, individual two-dimensional grid of cells: “Each element of the **two-dimensional grid** is referred to as a cell.” (’776 Patent at 1:33-34 (emphasis added)). **“Individual two-dimensional**

spreadsheets can be organized into a **larger** entity known as a notebook or workbook. The terms **spreadsheet** or worksheet will be used interchangeably herein, as will the terms notebook or workbook.” (*Id.* at 1:42-45 (emphases added)). When referring to multiple worksheets, the patent instead refers to “workbook”: “[A] workbook (which can contain multiple worksheets) . . .” (*Id.* at 11:29-31). Thus, in this patent, a “spreadsheet” is a single two-dimensional grid of cells.

Logical Unit of Calculation: The claim language ties “spreadsheet” to computations. Claim 1 says that a formula may be “in the spreadsheet” and that cell values in the spreadsheet are “computed.” The patent expressly defines its spreadsheet (and its workbook module) as a logical unit of computation: “the workbook module 115 can include **any logical unit of spreadsheet computation, comprising one or more spreadsheets** and associated formulas.” (’776 Patent at 11:38-41 (emphasis added)). “[T]he term **workbook can be substituted with this logical unit of spreadsheet computation** throughout the specification.” (*Id.* at 11:42-44 (emphasis added)). Thus, the patent expressly defines its “spreadsheet” as a single two-dimensional grid of cells forming a logical unit of computation. Microsoft’s proposed construction substitutes “calculation” for “computation” as being perhaps more jury-accessible, but either will do.

“Logical,” the term used in the patent to describe its workbook unit of computation, refers to an abstract, conceptual idea or construct, such as a formula or a grid of formulas, as distinct from a physical entity, such as a processor. For example:

Logical: 2. Conceptually true to a particular design or idea—for example, network transmissions travel in a circle around a logical ring, even though the ring shape itself is not physically apparent. *Compare* physical. (Ex. G (Microsoft Computer Dictionary (4th ed. 1999)) at 272).

The patent reinforces that its “spreadsheet” and “workbook” are logical, mathematical constructs, by expressly stating that its “workbook” is not limited to being a unit of computer-file storage or transfer. (’776 Patent at 11:28-44). Instead, the patent states, its workbook module “can include **any** logical unit of spreadsheet computation,” (*Id.* at 11:38-41 (emphasis added)). (*See also* Dkt. No. 28 at 11).

Each Cell Can Contain a Formula and Each Cell Can Contain a Data Value: Claim 1 refers to “a formula in the spreadsheet” and to “computed” “cell values in the spreadsheet.” The patent defines a spreadsheet cell as able to contain a formula for calculating a data value from another value (or values) or, if it does not hold a formula, it can contain a data value. (’776 Patent at 1:34-36).

“the spreadsheet”: The same spreadsheet that is referenced as “the spreadsheet” elsewhere in the claim, not a different copy of it.

After the first reference to “a spreadsheet” in the preamble of Claim 1, the claim language thereafter refers to “**the** spreadsheet.” It does not refer to “**a** spreadsheet” or “**a copy** of the spreadsheet” or “**an instance** of the spreadsheet.” By using the definite article “the,” Claims 1 and 10 refer to **the same spreadsheet** identified earlier in Claim 1. The patent specification confirms this construction, by using the word “copy” when it means to refer to a copy of a workbook. For example: “multiple copies of the same workbook to be loaded into memory simultaneously.” (’776 Patent at 1:58-59). By contrast, the asserted claims do not say “copy.” Similarly, the patent specification (including application claim 20 (Ex. E at NETVIEW 000200)) also refers to an “instance” of a spreadsheet, but the two asserted claims do not.

This construction is bolstered by standard patent claim-drafting practice, which calls for use of a definite article (e.g., “the” or “said”) to reference a previously recited element. (Ex. H (Robert C. Faber, *FABER ON MECHANICS OF PATENT CLAIM DRAFTING* (6th ed. 2010)) at § 10:7.4

at 10-43 (“When a previously identified element or step is repeated, it is introduced by a definite article ‘the’ or ‘said.’ ”)). *Cf. Baldwin Graphic Sys., Inc. v. Siebert, Inc.*, 512 F.3d 1338, 1342 (Fed. Cir. 2008). Claims 1 and 10 follow this standard claim-drafting practice, using the indefinite articles “a” and “an” to refer to newly introduced claim terms (e.g., “a spreadsheet” and “an output”) but using the definite article “the” to refer back to a previously recited item (e.g., “the spreadsheet” and “the computed cell values”).

Thus, Claim 1 requires that the spreadsheet referenced in the “computing cell values” step of Claim 1 must be the same spreadsheet referenced in the “associating” step of Claim 1, not merely a copy or instance of that spreadsheet.

2. “defining a parameter external to the spreadsheet”

<u>Context of Terms Used In Claim</u>	<u>Construction</u>
1. . . . defining a parameter external to the spreadsheet;	<p><u>“defining a parameter”</u>: Establishing the meaning of a parameter, including its identity and type, but not a specific value of the parameter.</p> <p><u>“parameter”</u>: A variable having an identity (e.g., name) and a type, but no specific value prior to run time.</p> <p><u>“a parameter external to the spreadsheet”</u>: A parameter existing and defined independently of the spreadsheet and its contents, such that the parameter’s definition does not refer to the spreadsheet and eliminating the entire spreadsheet does not affect the parameter.</p>

“defining a parameter”: Establishing the meaning of a parameter, including its identity and type, but not a specific value of the parameter.

Establishing the Meaning of a Parameter, Including its Identity and Type: The claim requires “defining” a parameter, not merely “naming” or “identifying” it. All parameter definitions in the patent designate both a name and a type. The patent’s Table 1, for example, establishes a name and a type for three parameters. (’776 Patent at 16:34-50). Similarly, the

patent describes an XML file that “defines” a workbook, and that file names two parameters and defines their types. (*Id.* at 10:40-65). The patent purports to incorporate by reference (*id.* at 1:8-12) the applicants’ July 13, 2001 provisional application (Ex. F). That application defined the alleged “invention” as associating a type with each parameter (and each result): “The **present invention associates type information with each workbook parameter** and result.” (Ex. F at 5 (emphasis added)). See *Akamai Techs., Inc. v. Limelight Networks, Inc.*, -- F.3d --, Nos. 2009-1372, -1380, -1416, -1417, 2010 WL 5151337, at *13-14 (Fed. Cir. Dec. 20, 2010) (claim construction requiring a feature was compelled by patent’s description of the “invention” as including that feature, even had the prosecution history suggested a broader construction).

But Not a Specific Value of the Parameter: “Defining” a parameter is unlike defining a constant value in that the former does not set a particular value for the parameter. The claim thus recites “defining a parameter” separately from “receiving a value for the parameter.” In particular, Claim 1 is clear that defining a parameter does not associate the parameter with a cell value (or cell) in the spreadsheet. Rather, there is a separate “associating” step in which the parameter is associated with the spreadsheet. Even then, the parameter is associated with the **entire** spreadsheet, not a particular cell or cell value.

If a parameter were “defined” as being a cell of the spreadsheet, then the only way to change the parameter’s value would be to change the value of that spreadsheet cell. But the patent seeks to set itself apart precisely by saying that this is not what it does. That is, the patent attempts to distinguish its alleged invention from certain “[p]rior art spreadsheet programs [that] make the user vary cell values if the user wants to perform different calculations.” (’776 Patent at 2:9-11). This also reinforces that an external parameter is not defined as a cell of the spreadsheet, as discussed below.

“parameter”: A variable having an identity (e.g., name) and a type, but no specific value prior to run time.

Having an Identity (e.g., Name): The claim says that a formula must be able to “reference the parameter,” which indicates that the parameter has an identity by which it can be referenced. The patent specification uses “parameter” to refer to a mathematical variable that has an identity—for example, the name “person” or name “period”—and a type, for example, “Person” and “Year” types. (’776 Patent at 10:60-63, 16:53-55). All parameters in the patent are variables having names.

Having a Type: The claims refer to “defining” a parameter, not just “naming” or “identifying” one. In this patent, a parameter thus has a type as well as an identity. A parameter’s type specifies the kind or category of values a parameter may have—for example, “integer” or “alphanumeric.” As noted, the applicants’ provisional patent application stated: **“The present invention associates type information with each workbook parameter and result.”** (Ex. F at 5 (emphasis added)). This history affords further support for Microsoft’s proposed construction of “parameter.” *Cf. Akamai Techs.*, 2010 WL 5151337, at *13-14 (Fed. Cir. Dec. 20, 2010); *O.I. Corp. v. Tekmar Co.*, 115 F.3d 1576, 1582 (Fed. Cir. 1997).

The rest of the patent is also consistent with this definition. All examples of “parameters” in the patent have a type. E.g., “This example file names the parameters 110 associated with the parameterized workbook [105] ‘person’ and ‘period’ **and defines them as ‘Person’ and ‘Year’ types, respectively.**” (’776 Patent at 10:60-63 (emphasis added); *see also id.* at 10:40-47, 16:37-40, 16:55). And: “In one embodiment, **the types associated with the parameters 110 and results 120** are described by an object-oriented type system[.]” (*Id.* at 9:25-27 (emphasis added)).

The patent ties this requirement that a parameter have a type to a supposed benefit of the alleged “invention.” (*See* ’776 Patent at 9:1-13). The patent touts that the same spreadsheet can be reused, without changing the spreadsheet, by changing the parameter’s definition, including its type. Type information, the patent notes, can “defin[e] a range of values” (*id.* at 2:52), which for the patent means that it can “constrain[] the set of legal data values the system can use for a parameter” (*id.* at 9:16-17). Thus, when a spreadsheet is reused in a different context, “[t]he system can use the type information to constrain the legal instantiations of a parameterized workbook.” (*Id.* at 10:7-8).

No Value Prior to Run Time: The claim separately recites, first, “defining a parameter” and, second, “receiving a value for the parameter at run time.” This separation indicates that the step of “defining” a parameter does not itself include providing any **value** for the parameter. Indeed, this is the difference between defining a parameter and a constant. A constant’s definition (e.g., the definition of the constant π) specifies its value. A parameter’s definition (e.g., the meaning of r , “radius”) does not. Thus, no parameter module in the patent specifies a specific value for a parameter. (*See, e.g.*, ’776 Patent at 16:34-50).

“a parameter external to the spreadsheet”: A parameter existing and defined independently of the spreadsheet and its contents, such that the parameter’s definition does not refer to the spreadsheet and eliminating the entire spreadsheet does not affect the parameter.

The claim’s reference to its parameter being “external to the spreadsheet” is another means by which the patent attempts to distinguish its invention from the conventional “what-if” usage of a spreadsheet, which changed the value associated with an internal spreadsheet cell in order to compute a new output value. Claim 1 indicates in at least three separate ways that a parameter must exist and be defined **independently of the spreadsheet and its contents**.

- First, the claim expressly states that the parameter is “external to the spreadsheet.”

- Second, the claim states that the “associating” step, not the parameter “defining” step, is what associates the parameter with the spreadsheet. In other words, “defining a parameter” does not associate the parameter with the spreadsheet in any way. It would be surplusage to have an “associating” step for two items that have already been associated. If the parameter definition referred to the spreadsheet, then the very definition of the parameter would associate the parameter with the spreadsheet—which is contrary to the claim language.
- Third, the claim specifies that the parameter is associated “with the spreadsheet” itself (rather than with a cell or other portion of the spreadsheet), and that the parameter “applies to the spreadsheet as a whole” (again, not just a cell or other portion of the spreadsheet).

Read together, this claim language recites in words what the patent application’s Fig. 1 illustrates. As depicted, the parameter module is logically separate and independent from the workbook module and its spreadsheet, such that the latter could be deleted without affecting the former, and such that the parameter is associated with the entire workbook module, not with some particular spreadsheet cell or other portion of the workbook module.

The patent’s text also supports this construction of the claim. The text refers to parameters as being “external to the spreadsheet.” (E.g., ’776 Patent at 3:22-23). The patent describes the consequence (and purpose) of this logical separation of the spreadsheet and the parameter and the output: that is, a user’s ability to change one module without affecting the other. “Parameterized spreadsheets . . . allowing computations to use varying inputs and allowing the output desired to be specified by the user without having to change the underlying spreadsheet 115.” (*Id.* at 12:28-35). “A creator . . . associates the parameter module 110 and the

results module 120 with the workbook module 115. **This association can facilitate the reuse of workbook calculations in different contexts.**” (*Id.* at 9:1-5 (emphasis added)). These different “contexts” may be reflected by different definitions of the parameters and/or outputs, without changing the spreadsheet or its contents. In other words, the parameter’s definition does not refer to or depend on the spreadsheet or its contents.

Further, as noted, the patent attempts to distinguish its “parameterized workbook” from standard “what-if” uses of a spreadsheet in which the input variable is a cell of the spreadsheet, whose value is varied in order to compute new results:

In existing spreadsheet programs, data values can be stored in cells either by user input (directly or through a user-input formula) or by the user associating the cells with an external data source, such as a query to a database. (’776 Patent at 1:62-65).

Prior art spreadsheet programs make the user vary cell values if the user wants to perform different calculations, including what-if scenarios. (*Id.* at 2:9-11).

With parameterized workbooks 105, a user **does not have to vary cell values** in order to perform different calculations, including what-if scenarios. (*Id.* at 11:45-47 (emphasis added)).

In sum, a person of skill in the relevant art reading this patent specification in 2002 would understand this claim language to refer to a parameter existing and defined independently of the spreadsheet and its contents, such that the parameter’s definition does not refer to the spreadsheet and eliminating the entire spreadsheet does not affect the parameter.

3. “design time”/“run time”

<u>Context of Terms Used In Claim</u>	<u>Construction</u>
1. . . . associating the parameter with the spreadsheet at design time to define a parameterized workbook, wherein the parameter applies to the spreadsheet as a whole, thereby allowing any formula in the spreadsheet to reference the parameter; receiving a value for the parameter at run time ;	“design time” : When the parameterized workbook is defined. “run time” : When the spreadsheet is used for computation.

“design time”: When the parameterized workbook is defined.

“run time”: When the spreadsheet is used for computation.

Claim 1 specifies that one step takes place at “design time” (“associating the parameter with the spreadsheet . . . to define a parameterized workbook”) and another at “run time” (“receiving a value for the parameter”). Thus, the claim uses “design time” to mean the point in time when the mathematical function is defined, not when it’s used, and “run time” to mean the later time when the function is used for computations.

The patent specification does not conflict with this reading as it does not use these claim terms at all. The Patent Office file does use these terms and, like Claim 1, ties them to the times when the parameterized workbook is defined and used, respectively. E.g., “the present invention allows this person to **create one parameterized workbook at design time**”; and “Once the parameterized workbook is created, **the present invention can, at run time, automatically compute and display** the ten commission reports” (Ex. E at NETVIEW 000070 (emphases added)).

The Patent Office file also clarifies that these terms are not shorthand for features supposedly unique to the alleged “invention” of the patent. Instead, the patent applicants applied these terms to the prior art as well, telling the Patent Office: “**As is common in the art, design time and run time** may be nearly coincident in certain embodiments of the present invention, for example to provide the user with immediate feedback at design time.” (*Id.* (emphasis added)).

4. **“a parameterized workbook”/“associating the parameter with the spreadsheet at design time to define a parameterized workbook”**

<u>Context of Terms Used In Claim</u>	<u>Construction</u>
1. . . . associating the parameter with the spreadsheet at design time to define a parameterized workbook, wherein the parameter applies to the spreadsheet as a whole, thereby allowing any formula in the spreadsheet to reference the parameter;	<p><u>“a parameterized workbook”</u>: A mathematical function for mapping defined inputs to defined outputs, and having three separate but associated logical modules: a parameter module defining a parameter, a workbook module defining a spreadsheet, and a results (outputs) module defining a named result having a type. Prior to run time, its parameter has no specific value.</p> <p><u>“associating the parameter with the spreadsheet at design time to define a parameterized workbook”</u>: At design time, associating the existing, defined external parameter with the spreadsheet. Until this step is performed, the parameter is not associated with the spreadsheet.</p>

The patent applicants gave self-contradictory definitions of the patent’s central term “parameterized workbook.” Specifically, they were internally inconsistent on whether or not this term requires a results module. Microsoft submits that this inconsistency on such a central term in the claim renders this claim language, and thus both asserted claims, indefinite and invalid under 35 U.S.C. § 112, ¶ 2, as explained in Section V, below. But, in the event the Court finds this claim language sufficiently capable of definition, Microsoft, in the alternative, proposes this construction:

“a parameterized workbook”: A mathematical function for mapping defined inputs to defined outputs, and having three separate but associated logical modules: a parameter module defining a parameter, a workbook module defining a spreadsheet, and a results (outputs) module defining a named result having a type. Prior to run time, its parameter has no specific value.

Mathematical Function “Mapping” Defined Inputs to Defined Outputs: The claim specifies that a parameterized workbook is defined by associating a defined external parameter with a spreadsheet. The claim does not expressly say that the defined parameterized workbook is then **used**, but it implies as much. Specifically, Claim 1 recites rendering an output based on

computed cell values in the spreadsheet of the parameterized workbook, based on a received value for the parameter. Thus, the claim implies that the defined parameterized workbook is used as a mathematical function to map the input parameter to the output.

The patent supports this reading of the claim language, by referring to its three-module “parameterized workbook” as a “function” and “computational building block” for computing defined result values from defined input parameter values. E.g.:

By associating parameters 110 and results 120 with each workbook 115, a single workbook 115 can perform a potentially unlimited number of **calculations**. This allows a parameterized workbook 105 to become a reusable **computational building block** in constructing a larger model 100. (’776 Patent at 9:9-13 (emphases added)).

In other words, **the parameterized workbook 105a is a function mapping a set of inputs 110 to a set of outputs 120**. (*Id.* at 13:19-21 (emphases added)).

Parameterized spreadsheets 105 **allow computations to be performed** and allow those computations to be influenced by external data sources through parameters, data ranges, and other mechanisms as described below. They provide increased flexibility to the user, allowing **computations to use varying inputs and allowing the output desired to be specified by the user without having to change the underlying spreadsheet 115**. (*Id.* at 12:28-35 (emphases added)).

As noted, the patent also refers to the workbook module of the parameterized workbook, as a “**logical** unit of spreadsheet computation.” (*Id.* at 11:38-44 (emphasis added)).

A person of ordinary skill in the relevant art reading this claim at the relevant time (2002) would have been familiar with the 1997 Clack and Braine paper quoted above, defining a “parameterized spreadsheet” in essentially the same manner:

A workbook has been defined as a collection of worksheets; we now allow one or more of these worksheets to take parameters and to return a result. Thus, the spreadsheet application is split into two parts: (i) simple worksheets that do not take parameters (and which provide a traditional non-reusable grid), and (ii) **worksheets that take one or more parameters and return a single result (which provide reusable ‘worksheet templates’)**. A parameterized worksheet is thus a function definition. . . . We believe that the grid format provides a useful structuring mechanism for the function body, wherein cells may contain

expressions with references to the worksheet's parameters. (Ex. A at MS-NV118405 (emphasis added)).

Parameter Module and Workbook Module: The claim language indicates that both a parameter and a spreadsheet exist (“defining a parameter external to the spreadsheet”) before they are associated with one another in the later “associating” step. Thus, the parameter and the spreadsheet are necessarily created as separate, non-associated entities. As shown in Fig. 1 and repeated throughout the patent, the parameterized workbook includes a parameter module, providing a name and type for at least one parameter, and a separate workbook module, defining the spreadsheet computational logic (e.g., formulas and data values). E.g., “The first parameterized workbook 105a includes a parameter module 110, a workbook module 115 and a results module 120.” (’776 Patent at 8:45-47). More specifically, the patent refers to the parameterized workbook as a “function” that “map[s]” inputs (values of a parameter defined in its parameter module) to outputs (values of an output defined in its results module). (*See id.* at 13:17-21).

Results Module: The claim language specifies that an output is rendered “based on” computed cell values, indicating that the “rendered output” is separate and distinct from those cell values. As shown in Fig. 1 and repeated throughout the patent, the parameterized workbook includes a “results module” naming and specifying a type for at least one result. For example, the purportedly incorporated provisional application defines its alleged “invention” as the association of a workbook with, *inter alia*, a list of typed results: “Parameterized spreadsheets. The present invention allows users to associate with each workbook a list of parameters (external inputs) and a list of results (outputs), thereby facilitating the reuse of workbook calculations in different contexts. The lists of parameters and results, and the types of these parameters and results, are recorded in a database.” (Ex. F at 4).

The patent repeatedly refers to this results module, or outputs module, as being a required module of the parameterized workbook:

As described above, a parameterized workbook 105a (FIG. 1) creates a set of outputs 120 (FIG. 1) based on a set of input parameters 110 (FIG. 1). In other words, **the parameterized workbook 105a is a function mapping a set of inputs 110 to a set of outputs 120.** ('776 Patent at 13:17-21 (emphasis added)).

The first **parameterized workbook 105a includes** a parameter module 110, a workbook module 115 and **a results module 120**. The parameter module 110 can include a list of one or more parameters (e.g., external inputs). The results module 120 can include a list of one or more results (e.g., outputs). . . . (*Id.* at 8:45-50 (emphases added)).

Parameterized spreadsheets 105 allow computations to be performed and allow those computations to be influenced by external data sources through parameters, data ranges, and other mechanisms as described below. They provide increased flexibility to the user, allowing computations to use varying inputs **and allowing the output desired to be specified by the user without having to change the underlying spreadsheet** 115. (*Id.* at 12:28-35 (emphasis added)).

By associating parameters 110 and **results 120 with each workbook 115**, a single workbook 115 can perform a potentially unlimited number of calculations. This allows a parameterized workbook 105 to become a reusable computational building block in constructing a larger model 100. (*Id.* at 9:9-13 (emphases added)).

No Parameter-Value Prior To Run Time: As noted above at 14, a parameter is an external variable, not a constant. As such, it receives a particular value during run time, not as part of the function's definition.

“associating the parameter with the spreadsheet at design time to define a parameterized workbook”: At design time, associating the existing, defined external parameter with the spreadsheet. Until this step is performed, the parameter is not associated with the spreadsheet.

As noted, the claim language “defining a parameter external to the spreadsheet” indicates that a parameter and a spreadsheet exist before they are associated with one another in the “associating” step; and thus until then are separate, non-associated entities. This is a defining characteristic of a mathematical function using a variable. For example, the equation defining the area of a circle, $A = \pi r^2$, does not exist until its constituent elements (input parameter r ,

constant π , named result *A*, and the multiplication and squaring operators) are associated together. Likewise, the parameterized workbook does not exist until its three constituent modules are associated together. This claim language reflects this, in part, by stating that the parameterized workbook is defined by associating the parameter with the spreadsheet. Both the parameter and the spreadsheet must already exist, of course, for them to be associated together. But, they cannot already be associated together for then this step would accomplish nothing. *See Bicon, Inc. v. The Straumann Co.*, 441 F.3d 945, 960 (Fed. Cir. 2006) (claim language cannot be superfluous).

5. **“reference the parameter”/“the parameter applies to the spreadsheet as a whole, thereby allowing any formula in the spreadsheet to reference the parameter” / “defining a formula within the spreadsheet at design time using (sic) parameter”**

<u>Context of Terms Used In Claim</u>	<u>Construction</u>
1. . . . associating the parameter with the spreadsheet at design time to define a parameterized workbook, wherein the parameter applies to the spreadsheet as a whole, thereby allowing any formula in the spreadsheet to reference the parameter;	<u>“reference the parameter”</u> : Bind directly to the external parameter without referring to a cell in the spreadsheet.
10. The method of claim 1, further comprising: defining a formula within the spreadsheet at design time using parameter	<u>“the parameter applies to the spreadsheet as a whole, thereby allowing any formula in the spreadsheet to reference the parameter”</u> : All cells of the spreadsheet and all formulas in the spreadsheet can reference the parameter. <u>“defining a formula within the spreadsheet at design time using (sic) parameter”</u> : When defining the parameterized workbook, defining a spreadsheet formula that references by name the external parameter (not a cell of the spreadsheet) for calculating a data value from another value.

“reference the parameter”: Bind directly to the external parameter without referring to a cell in the spreadsheet.

The claim specifies that the external parameter “applies to the spreadsheet as a whole” such that “any formula in the spreadsheet” can “reference the parameter.” This language

reinforces that the parameter is not a part of the spreadsheet (e.g., a cell or cell value), and that formulas refer to the parameter itself, not to a cell of the spreadsheet.

The patent acknowledges that formulas in prior-art spreadsheets could reference other cells in the spreadsheet: “Formulas stored in worksheet cells can reference other cells that are in worksheets in the same workbook[.]” (’776 Patent at 1:50-52). But, the patent criticizes some of the prior art for requiring that cell values be changed to perform a new calculation: “Prior art spreadsheet programs make the user vary cell values if the user wants to perform different calculations, including what-if scenarios.” (*Id.* at 2:9-11). Unlike this “what-if scenario” prior art, the parameterized workbook’s formula recited in the claims refers not to cells of the spreadsheet itself but instead to a parameter external of the spreadsheet. (E.g., ’776 Patent at 11:1-3 (“the spreadsheet . . . contains formulas that use the parameters person and period in them”)).

The patent applicants made this point to overcome a Patent Office rejection of their application claims over a prior-art patent to Jamshidi of IBM. The applicants distinguished Jamshidi as disclosing the binding of a parameter to a particular spreadsheet cell, such that a spreadsheet formula would need to reference that cell rather than directly reference the external parameter: “[In Jamshidi] [t]he values for the data binding parameters are specified by formulas in the spreadsheet, but **data values returned from the data source are bound to cells in the spreadsheet and not to formulas.** This is a consistent and important distinction and it is further indication that Jamshidi is disclosing a method for a parameterized data binding, as is known in the art, and not a parameterized workbook. **In the present invention, any formula in the spreadsheet can reference the workbook parameters because they apply to the spreadsheet as a whole.**” (Ex. E at NETVIEW 000076-77 (emphases added)). Thus, the applicants declared

that the claim language meant that parameters were bound to formulas and not to spreadsheet cells, and thus that formulas referred directly to external parameters and not to spreadsheet cells, as in this prior art.

If the Court concludes that the word “bind” would confuse a jury, then “refer” can be substituted without losing much precision.

“the parameter applies to the spreadsheet as a whole, thereby allowing any formula in the spreadsheet to reference the parameter”: All cells of the spreadsheet and all formulas in the spreadsheet can reference the parameter.

The claim language says that “the parameter applies to the spreadsheet as a whole.” As the patent defines a spreadsheet as comprising a grid of cells, by using the phrase “spreadsheet as a whole” the claim must be understood to mean that a parameter can be referenced by all of the spreadsheet’s cells. And since any cell of a spreadsheet can hold a formula, by saying that “any formula in the spreadsheet” can reference the external parameter, the claim likewise must mean that any and all cells of the spreadsheet can reference the parameter.

“defining a formula within the spreadsheet at design time using the parameter”: When defining the parameterized workbook, defining a spreadsheet formula that references by name the external parameter (not a cell of the spreadsheet) for calculating a data value from another value.

This claim language is in dependent Claim 10, which recites that the formula is “using” the parameter. As discussed above in connection with Claim 1, any cell of the spreadsheet must be able to reference the external parameter by name—and to do so directly, not via another cell of the spreadsheet. The language of Claim 10 simply adds the step of actually defining such a formula at design time, for calculating a value from another value (or values). The patent describes formulas using a parameter by referring to the parameter by name. (E.g., ’776 Patent at 20:56-63).

6. **“receiving a value for the parameter at run time”**

<u>Context of Terms Used In Claim</u>	<u>Construction</u>
1. . . . receiving a value for the parameter at run time;	“receiving a value for the parameter at run time” : Receiving a value for the external parameter. This step happens at run time not at design time. This step cannot be varying the value of a spreadsheet cell.

“receiving a value for the parameter at run time”: Receiving a value for the external parameter. This step happens at run time not at design time. This step cannot be varying the value of a spreadsheet cell.

Run Time Not Design Time: As noted, the claim language identifies the associating step as occurring at design time, and another step, this “receiving” step, as occurring at run time. As with any variable input to a mathematical function (e.g., a value for the radius when calculating the area of a circle), the value of the input parameter is received when the function is used for computation, not as part of defining the function itself. Thus, the patent describes a parameter value being received when calculations are to be performed: “In another embodiment, the system further comprises **a calculation module to receive a value for a first parameter within the list of parameters and generate a first result based at least in part on the value and the spreadsheet.**” (’776 Patent at 3:29-33 (emphasis added)).

Not Varying Cell Value: The claim does not expressly state how a parameter value is received, but it is clear that the computation recited in the claim does not result from **varying** (even temporarily) **the value associated with a cell** of the spreadsheet. The claim specifies that a parameter “applies to the spreadsheet as a whole.” Again, this precludes applying the parameter to only a portion of the spreadsheet, such as a single cell. And the claim specifies associating a parameter “with the spreadsheet.” This too is different from associating a parameter with a spreadsheet cell or other portion of the spreadsheet. Finally, the claim refers only once to cell values, when it discusses “computing cell values in the spreadsheet,” here again

something different from receiving a new input data value for a spreadsheet cell as in the prior art the applicants purported to distinguish.

The patent reinforces this reading of the claim. As noted, the patent criticizes a type of conventional spreadsheet usage in which calculations are performed by providing a new value for a cell of the spreadsheet: “Prior art spreadsheet programs make the user vary cell values if the user wants to perform different calculations, including what-if scenarios.” (’776 Patent at 2:9-11). And, the patent distinguished such what-if scenarios even if the new cell value was provided only temporarily, without storing it. (*Id.* at 1:39-41 (“perform what-if scenarios only by modifying cell values and **viewing or saving** the effects of the changes”) (emphasis added)). In contrast, the patent proclaims, “With parameterized workbooks 105, **a user does not have to vary cell values in order to perform different calculations**, including what-if scenarios.” (*Id.* at 11:45-47 (emphasis added)). This express distinction over a prior art approach confirms that this step of receiving a value for the parameter cannot be varying the value of a spreadsheet cell, not even temporarily for viewing purposes. *See Ormco Corp. v. Align Tech., Inc.*, 498 F.3d 1307, 1313-14 (Fed. Cir. 2007).

7. “rendering an output based on computed cell values”

<u>Context of Terms Used In Claim</u>	<u>Construction</u>
1. . . . receiving a value for the parameter at run time; computing cell values in the spreadsheet that are dependent, directly or indirectly, on parameter; and rendering an output based on the computed cell values.	“rendering an output based on computed cell values” : Based on the two or more computed cell values, providing a value for a named result (output) of the parameterized workbook.

“rendering an output based on computed cell values”: Based on the two or more computed cell values, providing a value for a named result (output) of the parameterized workbook.

Two or More Computed Cell Values: The claim says “cell **values**,”—that is, using the plural.

Provide Value for Named Result of the Parameterized Workbook: In the patent, the “output” of the computation is a data value of the result defined in the results module of the parameterized workbook. (*See, e.g.,* ’776 Patent at 12:5-23, 13:17-25).

8. Claims As A Whole

No Particular Machine: The claim does not require any particular machine to perform any of its steps.

Mathematical Operation: The claim is directed to a mathematical operation.

As briefed on Microsoft’s pending motion for summary judgment under Section 101 of the Patent Act (Dkt. No. 27), Claims 1 and 10 are directed to the mathematical operation of defining and using a function (“parameterized workbook”) to perform computations, and do not restrict who or what is used to define or use that function. These claims do not require (or preclude) any particular machine to perform any of its steps. The patent states that its “parameterized workbook” is **not** necessarily implemented as software program instructions running on a computer: “The modules throughout the specification can be implemented as a software program (e.g., a set and/or a sub-set of processor instructions and the like) and/or a hardware device (e.g., ASIC, FPGA, processor, memory, storage device and the like).” (’776 Patent at 8:63-67).

Non-Functional Claim Language: The claim recites only information, abstract operations, and intended uses that do not transform any physical article or affect how any physical process or device functions. For example, the term “parameter” refers to information and implies that its value is intended to be varied. But, the claim does not positively recite any step that varies the value of the parameter. Thus, this variability characteristic is a mere intended use.

Not only do these claims fail to recite any physical mechanisms, they fail to recite anything that affects how any physical mechanism operates. These claims recite only information (e.g., values, formulas, spreadsheet), abstract operations (e.g., computing), and implied intended uses (e.g., parameter), without linking any of this abstract information to any

physical substrate, machine or article. Thus, nothing recited in the claims transforms any recited physical article or affects how any recited physical process or device functions. For this reason, none of this claim language has any patentable “weight” or consequence when comparing the claim to the prior art. *See generally In re Nehls*, No. 2007-1823, 2008 WL 258370, at *10 (B.P.A.I. Jan. 28, 2008) (precedential) (expanded panel); *Astrazeneca LP v. Apotex, Inc.*, -- F.3d --, Nos. 2009-1381, -1424, 2010 WL 4286284, at *21 (Fed. Cir. Nov. 1, 2010).

For example, the claim’s “parameter” is a variable, whose value is **intended** to be changed (unlike a “constant”). But, neither asserted claim positively recites changing the value of this “parameter.” Rather, they recite a **single** step of receiving a value for the parameter, which is true of a constant as well. In other words, the temporal span of the asserted claims’ steps begins with defining the external parameter and ends with rendering an output based on a computation on a single receipt of a parameter value. The claims simply do not speak to (or require anything of) the time period after that single output is rendered. Thus, the variability of a parameter is merely an intended use in this claim, having no functional impact on anything recited in the claim.

Each step of Claim 1 (defining, associating, receiving, computing, and rendering) likewise constitutes “non-functional descriptive material.” *Cf. In re Nehls*, No. 2007-1823, 2008 WL 258370, at *10 (B.P.A.I. Jan. 28, 2008). None of these steps positively recites performing a manipulation of anything that is not abstract. None of these steps changes or affects the way any positively recited manipulative method step is performed or how any device performing such step functions, such as the efficiency, accuracy or other functional characteristic of the step.

IV. **SYSTEMATIC ERRORS IN PLAINTIFF'S PROPOSED CONSTRUCTIONS**

Plaintiff's draft proposed constructions are more complicated than the claim language they purport to construe. They introduce unexplained terms that will not be readily understood by a jury, such as "formally declaring," "execution instances," "persistent spreadsheet model," "programmable function," and "functionalized spreadsheet." Plaintiff also repeatedly rewrites the claims to include supposed limitations that are not in the claims themselves, including limitations as to what or who performs a recited step, how it is performed, and what happens before the method begins, and after it ends. Four examples of this improper rewriting of the claims will illustrate the problem.

1. Plaintiff reads into the claims a supposed limitation that the spreadsheet be formed by a "conventional spreadsheet program." But the asserted claims say nothing about what creates the spreadsheet. The claims are directed to a method "of utilizing a spreadsheet" and merely assume the existence of a spreadsheet. They do not recite creating the spreadsheet or how that came about. Nor do the asserted claims refer to a spreadsheet program, conventional or otherwise, or even to a computer.

2. Plaintiff reads into the asserted claims supposed limitations that the spreadsheet be stored "persistently." But asserted claims 1 and 10 recite nothing about storing persistently any information. On the contrary, these claims recite a **single** "computing cell values" step. Nothing in the claim requires the "spreadsheet" (or anything else) to persist after that computing step is completed. As noted, the temporal span of the claims' recited steps begins with defining the external parameter and ends with rendering an output based on a computation on a single receipt of a parameter value. The claims simply do not speak to the time periods before or after this limited time period. *Cf. Ormco*, 498 F.3d at 1317 (specification and prosecution history

disclaimers limiting scope of a step recited in other claims do not limit claims directed to stage earlier than that step).

3. Plaintiff reads into the asserted claims supposed limitations about multiple uses of the spreadsheet by multiple users and programs. But, the claims say nothing of the kind. Again, Claim 1 recites only a single “computing cell values” step, and does not require multiple uses of the spreadsheet. Likewise, again, the claim does not recite **any** computer or computer program, much less multiple programs.

4. Plaintiff reads into the asserted claims supposed limitations about computer programs and computer program “environments.” But once again, nothing in the claims requires a computer or computer program. Indeed, to the contrary, the patent states that **none** of the three modules of the “parameterized workbook” need to be implemented as software. (’776 Patent at 8:63-67).

If NetView wanted such limitations in the claims, it had the opportunity to add them to the claims for examination by the Patent Office. But once the patent issues, the public is entitled to rely on the language of the claims as granted. Claim construction is not a vehicle for wishful rewriting of the claims issued by the Patent Office. *See SmithKline Beecham*, 403 F.3d at 1339-40.

V. THE CLAIMS ARE INDEFINITE

As noted, the concept of the patent’s “parameterized workbook” was old at the time of the subject application, and had been described in the prior art by, for example, Clack and Braine who used similar terminology—“parameterized worksheet.” Nevertheless, the exact claim term “parameterized workbook” did not have a single, fixed meaning in the art in 2002. For example, a search of “parameterized workbook” in the Patent Office’s database of issued U.S. patents returns only the ’776 Patent. Therefore, it was especially incumbent upon the patent applicants

to give this claim term a clear, precise meaning. *Cf. J.T. Eaton & Co. v. Atlantic Paste & Glue Co.*, 106 F.3d 1563, 1570 (Fed. Cir. 1997) (“the dispositive claim limitation is a term unknown to those of ordinary skill in the art at the time the patent application was filed. It thus fell to the applicants, as a duty, to provide a precise definition for the 120°F limitation”).

This they did not do. On the contrary, the patent applicants gave the public inconsistent definitions for this claim term. This failure renders these asserted patent claims invalid under Section 112, ¶ 2 (“The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.”).

The goal of our Patent System is to encourage innovation. Patent claims with unclear boundaries have the opposite effect:

The statutory requirement of particularity and distinctness in claims is met only when they clearly distinguish what is claimed from what went before in the art and clearly circumscribe what is foreclosed from future enterprise. A zone of uncertainty which enterprise and experimentation may enter only at the risk of infringement claims would discourage invention only a little less than unequivocal foreclosure of the field.

United Carbon Co. v. Binney & Smith Co., 317 U.S. 228, 236 (1942) (applying predecessor statute).

As quoted above (*see* pg. 19), the patent application repeatedly defined “parameterized workbook” as having three modules, including a results (outputs) module as shown in Fig. 1. The patent describes no “parameterized workbook” lacking a results module. But, in their public correspondence with the Patent Office, the patent applicants contradicted themselves on this point and said that “parameterized workbook” is a **two**-module construct, with only a parameter module and workbook module: “In the present invention, a parameterized workbook is composed of a spreadsheet and one or more parameters, the parameters being associated with the spreadsheet and referenced by formulas in the spreadsheet.” (Ex. E at NETVIEW 000072). This

is inconsistent with Fig. 1 of the patent and the patent text which says, repeatedly, that the parameterized workbook is composed of three modules, including a results module.

A claim satisfies the definiteness requirement “[i]f one skilled in the art would understand the bounds of the claim when read in light of the specification.” *Exxon Research & Eng’g Co. v. United States*, 265 F.3d 1371, 1375 (Fed. Cir. 2001). Under current Federal Circuit precedent, a patent claim will be found indefinite if it “is insolubly ambiguous, and no narrowing construction can properly be adopted.” *Id.* Here, the applicants contradicted themselves defining this central term in the claim, “parameterized workbook,” a term, moreover, with no established meaning in the art. Thus, a person of skill in the art was left guessing as to which, conflicting, definition is correct and which is not, rendering the claims indefinite.³

Dated: January 10, 2011

Respectfully submitted,

MICROSOFT CORPORATION,
By Its Attorneys,

s/John D. Vandenberg

William L. Boesch (BBO No. 558742)

boesch@srbc.com

SUGARMAN, ROGERS, BARSHAK & COHEN, P.C.

101 Merrimac Street, 9th Floor

Boston, Massachusetts 02114

Telephone: (617) 227-3030

Facsimile: (617) 523-4001

³ To the extent that the Court adopts Plaintiff’s proposed constructions, Claims 1 and 10 may fail the claiming requirements of Section 112, ¶ 2 in other ways as well. *See* Dkt. No. 39 at 24-29.

Kristin L. Cleveland (admitted *pro hac vice*)

kristin.cleveland@klarquist.com

Klaus H. Hamm (admitted *pro hac vice*)

klaus.hamm@klarquist.com

John D. Vandenberg (admitted *pro hac vice*)

john.vandenberg@klarquist.com

KLARQUIST SPARKMAN, LLP

121 S.W. Salmon Street, Suite 1600

Portland, Oregon 97204

Telephone: (503) 595-5300

Facsimile: (503) 595-5301

CERTIFICATE OF SERVICE

I certify that on January 10, 2011, DEFENDANT MICOROSOFT CORPORATION'S OPENING BRIEF IN SUPPORT OF ITS PROPOSED CLAIM CONSTRUCTIONS was filed through the ECF system and will be sent electronically to the registered participants as identified on the Notice of Electronic Filing (NEF) and paper copies will be sent to those indicated as non-registered participants.

s/John D. Vandenberg
John D. Vandenberg